

MENU

SEARCH

INDEX

DETAIL

JAPANESE

LEGAL  
STATUS

1 / 1

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : **05-108730**  
 (43)Date of publication of application : **30.04.1993**

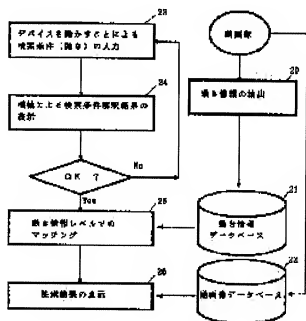
(51)Int.Cl. **G06F 15/40**  
**G06F 12/00**

(21)Application number : **03-242329** (71) **INTERNATL BUSINESS MACH**  
**CORP <IBM>**  
 (22)Date of filing : **29.08.1991** (72) **IOKA MIKIHIRO**  
 (73) **KUROKAWA MASAHIRO**

## (54) RETRIEVAL FOR ANIMATION IMAGE DATA BASE

## (57)Abstract:

PURPOSE: To improve the degree of freedom of retrieval by inputting movement intended by a user as a retrieving condition in retrieving an animation image.  
 CONSTITUTION: A movement information data base 21 is constructed, based upon a motion vector string extracted from a moving image. A user inputs a vector string to be a retrieving condition by moving a device such as a mouse capable of changing a spatial position (step 23). Then matching is executed and a scene including the inputted movement is retrieved (step 25). Thereby it is unnecessary for the user to describe movement to be the retrieving condition by language or a conditional-expression.





特開平5-108730

(43) 公開日 平成5年(1993)4月30日

(51) Int.Cl. <sup>5</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 15/40	5 3 0 Q	7060-5L		
12/00	5 4 7 D	7832-5B		

審査請求 有 請求項の数15(全 20 頁)

(21) 出願番号	特願平3-242329	(71) 出願人	390009531 インターナショナル・ビジネス・マシーンズ・コーポレイション INTERNATIONAL BUSINESS MACHINES CORPORATION アメリカ合衆国10504、ニューヨーク州 アーモンク (番地なし)
(22) 出願日	平成3年(1991)8月29日	(72) 発明者	井 岡 幹 博 東京都杉並区上井草3-14-15
		(72) 発明者	黒 川 雅 人 神奈川県川崎市宮前区鷺沼2-10-53
		(74) 代理人	弁理士 頓宮 孝一 (外4名)

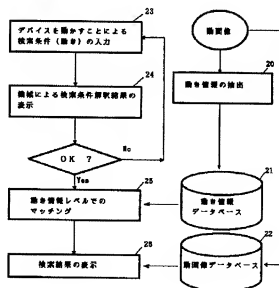
(54) 【発明の名称】 動画像データベースの検索

## (57) 【要約】

【目的】 本発明の目的は、動画像の検索を行なう際に、利用者が意図した動きを検索条件として入力することを可能にし、もって検索の自由度を高めることにある。

【構成】 動画像から抽出された動きベクトル列に基づいて、動き情報データベース21が構築される。利用者は、マウス等の空間的な位置を変化させることの可能なデバイスを動かして、検索条件となるベクトル列を入力する(ステップ23)。マッチングが実行されて、入力された動きを含むシーンが検索される(ステップ25)。したがって、利用者は、検索条件となる動きを言葉や条件式で記述しなくてよい。

## ( 発 明 )



【特許請求の範囲】

【請求項1】(a)動画像を蓄積する手段と、

(b)上記蓄積された動画像中に登場する物体の動きのデータを、上記蓄積された動画像の該動きを含む部分にアクセスするためのデータとともに蓄積する手段と、

(c)空間的な位置を変化させることの可能な、所望の動きを入力するためのデバイスと、

(d)上記デバイスの位置の変化に応じて、動きのデータを生成する手段と、

(e)上記手段(d)の生成した動きのデータと上記手段

(b)に蓄積された動きのデータとのマッチングを行う手段

を含む、動画像蓄積・検索システム。

【請求項2】生成された動きのデータ対話式に修正する手段を含む、請求項1記載のシステム。

【請求項3】上記手段(e)によって入力した動きに類似すると判定された動きを含む部分の動画像にアクセスし、表示するための手段を含む、請求項1記載のシステム。

【請求項4】蓄積された動画像からその一部分を検索するためのシステムであって、

(a)上記蓄積された動画像中に登場する物体の動きのデータを蓄積するデータベースと、

(b)空間的な位置を変化させることの可能な、所望の動きを入力するためのデバイスと、

(c)上記デバイス(b)の空間的な位置の変化に応じて、動きのデータを生成する手段と、

(d)上記手段(c)の生成した動きのデータと上記動き情報データベースに蓄積された動きのデータとのマッチングを行う手段

を含む、動画像検索システム。

【請求項5】1つの動きに関して蓄積または生成されるデータは、動きの開始時刻、動きの継続時間、画面上での動きの開始点の座標及び各々が所定時間内の動きを表わすベクトルの時系列である、請求項4記載のシステム。

【請求項6】上記手段(d)は、入力した動きの曖昧さに関する指示に応じて、マッチングの手法を選択する、請求項4記載のシステム。

【請求項7】部分動画像に含まれる複数の動きの開始時刻の前後関係を記述するファイルと、複数の動きが入力されたことに応じて、上記ファイルにアクセスし、開始時刻の前後関係が入力された動き相互の前後関係と同じである同一部分動画像内の複数の動きを検索する手段を含む、請求項5記載のシステム。

【請求項8】蓄積された動きの開始点が、画面上を分割して設定された複数の領域のうちのどれに存在するかを記述するファイルと、動きが入力されたことに応じて、上記ファイルにアクセスし、開始点の存在する領域が同じである蓄積された動きを検索する手段を含む請求項5

記載のシステム。

【請求項9】蓄積された動画像中に登場する物体の動きのデータを、該動きを含む部分の動画像にアクセスするためのデータとともに蓄積する手段と、空間的な位置を変化させることの可能な、所望の動きを入力するためのデバイスを備えた動画像処理システムにおいて、所望の部分を検索するための方法であって、

(a)上記デバイスの位置の変化に応じて、動きのデータを生成するステップ、

(b)上記生成された動きのデータと上記蓄積手段に蓄積された動きのデータとのマッチングを行うステップ、及び

(c)上記ステップ(b)で入力した動きに類似すると判定された動きを含む部分の動画像にアクセスし、表示するステップ

を含む、動画像検索方法。

【請求項10】蓄積された動画像から所望の部分を検索するために、動きを検索条件として入力するためのシステムであって、

(a)動きのデータを入力するための装置と、

(b)スクロールバーの先端位置を動きの開始時刻に、スクロールバーの長さを動きの継続時間にそれぞれ対応させて、入力された動きを表わすスクロールバーを表示する手段と、

(c)スクロールバーの位置または長さを修正する手段と、

(d)スクロールバーの位置または長さを修正に応じて、対応する動きのデータを修正する手段

を含む、入力システム。

【請求項11】動きを再生する手段をさらに含む、請求項10記載のシステム。

【請求項12】フレーム列によって構成される動画像からそれに含まれる動きの情報抽出する方法であって、

(a)動画像を所定の基準にしたがってオリジナルのフレーム列よりも短いフレーム列である部分動画像に分割し、

(b)部分動画像の各々について、

(b1)その開始フレームを複数のブロックに分割し、

(b2)分割されたブロックの各々について、現在のフレームと直前のフレームを比較して、ブロック間の誤差が最小になるベクトルを探索し、そのようなベクトルが発見されたなら、直前のフレームにおける当該ブロックの画面上の位置と発見されたベクトルとから現在のフレームにおける当該ブロックの画面上の位置を算出することを反復する

ことを特徴とする方法。

【請求項13】所定フレームおきに、画面上でのブロックの分布を調べ、ブロックの空白領域があれば、該空白領域にてブロックを新しく生成するステップを含む、請求項12記載の方法。

3

【請求項14】 フレーム列によって構成される動画像に登場する物体の動きの情報を蓄積する動き情報データベースを構築するための方法であって、

(a) 動画像を所定の基準にしたがってオリジナルのフレーム列よりも短いフレーム列である部分動画像に分割し、

(b) 部分動画像の各々について、

(b1) その開始フレームを複数のブロックに分割し、

(b2) 分割されたブロックの各々について、現在のフレームと直前のフレームを比較して、ブロック間の誤差を最小にする当該ブロックの動きベクトルを探索し、動きベクトルが発見されたなら、直前のフレームにおける当該ブロックの位置と該動きベクトルとから現在のフレームにおける当該ブロックの位置を算出することを反復して、ブロック毎に動きベクトル列を生成し、

(b3) 類似する挙動を示す動きベクトル列を1つの動きベクトル列に統合し、

(b4) 上記ステップ(b3)の結果生成されたベクトル列を当該部分動画像の識別子と関連づけて記憶装置に蓄積する

ことを特徴とする方法。

【請求項15】 上記ステップ(b3)において、生成されたベクトル列をフレームのサンプリング間隔よりも長い時間間隔でサンプリングすることを特徴とする、請求項14記載の方法。

【0001】

【発明の詳細な説明】

【産業上の利用分野】 本発明は、利用者がマウス等のデバイスの空間的な位置を変化させて入力した動きの情報を検索条件として用いて、動画像の一部分を検索するシステム及び方法に関する。

【0002】

【従来の技術】 従来、蓄積された動画像の中からその一部分であるところのシーンを検索するための方式として、図1に示す如く、検索条件をキーワードで記述する方法が知られている。その方法によれば、ある動画像(フレーム列)を動画像データベース12に蓄積する一方、キーワードを使った検索を念頭において、各シーンの内容を記述し(ステップ10)、そのシーンの格納アドレスとともに、内容の記述を予め文字数値データベース11に登録しておく。検索時には、キーワードを含む検索条件を入力し(ステップ13)、キーワードと文字数値データベース11に蓄えられた記述とのマッチングが実行される(ステップ14)。マッチしたシーンの表示は、動画像データベース12の、当該シーンの格納アドレスをアクセスすることによって行われる(ステップ15)。しかしながら、この方法には、以下に述べるような問題点が存在する。

【0003】 まず、人間が動画像を見て、内容を言葉で記述しなければならない。これは、多大な労力を要す

4

る。また、すべての動画像を1人の人間が見るわけではないので、内容記述に個人差ができ、内容の不偏性及び客観性において問題がある。さらに、利用者は、内容記述者でない場合が一般であるので、利用者の要求とデータベースの記述が一致しない場合が多く存在する。これは、検索のヒット率の低下を招く。

【0004】 そこで、シーンの内容記述に代えて、動画像に登場する物体の動き情報に対して検索を試みる研究がなされている。特開平1-224875号公報の動画像情報検索方式では、動画像に登場する物体の物理的な状態をデータベースに蓄積し、特定の物体の状態変化を記述する論理的な条件式を使って検索を試みようとするものである。物体の状態変化とは、具体的には物体の2次元座標の時系列であり、それが動き情報データベースに蓄積される。利用者は、検索条件を言葉で表現すればよく、検索条件は自動的に条件式に変換され、条件式に合致する状態変化をする物体を含むシーンが検索される。しかしながら、利用者は物体の動きを言葉または論理的な条件式で表現しなければならないが、それは困難

である。

【0005】 例えば、時刻t1に画面上にある座標(x1, y1)にあった物体Sが時刻t2に座標(x2, y2)へ曲線を描いて移動したシーンを、精度よく検索したい場合には、2つの時刻での時間、位置及び物体がその間を移動したという条件を入力するだけでは足りない。移動途中の位置や速さも表現しなければならないが、それを利用者が文字数値で表現しつくすことはほとんど不可能である。

【0006】 また、この公報の方式では、物体の名前を検索条件において特定しなければならない。したがって、物体が登録されている名前を正確に知らなければ、検索に失敗するという問題点もある。

【0007】 さらに、この公報では、動き情報として、画面上に登場する物体の2次元座標をフレームごとに抽出することが提案されているけれども、自動抽出の手法は提示されていない。たとえ、検索者が所望の動きを計算機に伝える環境が整ったとしても、検索の対象となる動きの情報を動画像から抽出するのに多大な労力をかけていたのでは、実用的であるとはいえない。

【0008】

【発明が解決しようとする課題】 したがって、本発明の目的は、動画像の検索を行なう際に、利用者が意図した動きを検索条件として入力することを可能にし、もって検索の自由度を高めることである。本発明の他の目的は、物体の名前を知らなくても、動きを条件とする検索を可能にすることである。本発明のさらに他の目的は、動きを条件とする検索の前提となる動き情報データベースの構築に自動化手法を導入することによって、人間の労力を軽減するとともに、蓄積される動き情報の内容の客観性を確保することである。

【0009】

5

【課題を解決するための手段】図2は、本発明の処理とデータの流れを示したものである。動画画像は、動画データベース22に蓄積される一方、画像処理によって動画画像から物体の動きの情報が抽出され、動き情報データベース21に蓄積される。

【0010】動き情報の抽出(ステップ20)は、次の3つの段階A、B、Cからなる。

A: フレーム列で構成される動画画像を、所定の基準にしたがって、オリジナルよりも短いフレーム列である部分動画画像に分割する。シーンチェンジを分割の基準とするのが一般的である。

【0011】次のB、Cは、部分動画画像ごとに実行される。ステップCは省略してもよいけれども、検索の効率を考えると、これを実行するのが好ましい。

【0012】B: まず、画面をブロックに分割し、その動きを追跡することによって、動きベクトル列を発生させる。次に、類似した挙動を示す動きベクトル列を統合することによって、物体の動きを代表するベクトル列を自動的に生成する。1つの代表動きベクトル列につき生成されるデータは、フレーム間変位(代表動きベクトル)の時系列のほかに、動きの開始フレームの番号、動きが継続しているフレームの数及び開始フレームでの動きの開始点の座標値が含まれる。

【0013】C: 次に、代表動きベクトル列データを、フレームのサンプリング間隔よりも長い時間間隔でサンプリングして、動きインデックスを生成する。1つの動きインデックスについて生成されるデータは、各々が当該サンプリング間隔内の動きを表すベクトルの時系列のほかに、動きの開始時間、動きの継続時間及び動きの開始点の座標値が含まれる。

【0014】部分動画画像の識別子、動画データベース22における該部分動画画像の格納場所を示すデータ及び動きインデックスの3者を関連づけて記憶装置に蓄積することによって、動き情報データベースが構築される。

【0015】動きを条件とした検索を実行するために、利用者がその空間的な位置を変化させることのできるデバイスを使用する。利用者は任意の位置から別の任意の位置まで該デバイスを移動させることによって、所望の動きの始点の位置、終点の位置、移動中の位置、及び要した時間を検索条件として入力する(ステップ23)。動く物体の名前を入力する必要はない。計算機に入力された条件は表示装置のスクリーンに表示される(ステップ24)。これによって、条件の確認、条件の再入力、編集を行うことが可能になり、より自由度の高い検索を行うことができる。入力された条件と動き情報データベース21に蓄えられた動きのデータとの間でマッチングが実行され、各シーンの入力条件との類似度が判定される(ステップ25)。その結果、類似度の高いシーンから\*

$$\sum (S_i(x-dx, y-dy) - S_{i-1}(x, y))^{**2}$$

ここで、x, yは画面の座標であり、 $S_{i-1}(x, y)$ は第i-1フ

6

\*順に表示される(ステップ26)。

【0016】

【実施例】

1. 動き情報の抽出及び動き情報データベースへの蓄積  
以下では、図2のステップ20の詳細を、図3に示す処理の流れにしたがって説明する。

【0017】a) 動画画像の入力(ステップ31)

まず、動画画像を、蓄積媒体から処理装置に入力する。動画画像がアナログ信号で蓄積されているならば、予めそれをデジタル信号に変換する処理が必要である。

【0018】動画画像の動画データベース22への蓄積方法は、本発明の要旨に関係しないので、詳細な説明は省略する。ただ、動画データベース22がどのような記憶装置を使ってどのように構築されようとも、本発明は実施可能であることに留意されたい。画像データのみならず、絵に同期している音声データ、時間データなどが動画データベース22に含まれていて差し支えないし、データが適当に圧縮されていて差し支えない。

【0019】実施例では、動画データベース22に蓄積されている任意のフレームの画像データにアクセスするときに、フレーム番号から動画データベース22における当該フレーム画像の格納アドレスを生成する。フレーム番号から格納アドレスへの変換方法は、表を用いてもよいし、特願平3-156214号に開示された方法を使って、フレーム番号から画像データの格納アドレスを算出してもよい。

【0020】b) シーンチェンジの検出(ステップ32)  
次に、動画画像のシーンチェンジを検出する処理が実行され、各シーンの開始フレーム番号及びフレーム数が出力される。本実施例では、シーンチェンジから次のシーンチェンジまでを1つの期間として、期間ごとに物体の移動軌跡を求める。シーンチェンジを検出する技術としては、例えば特開平3-73076号公報に開示されているような、公知の技術を用いることができる。

【0021】c) 動きベクトル列の生成(ステップ33)  
以下の処理は、シーン毎に実行される。まず、図4に示すように、処理対象のシーン(注目シーン)の開始フレームを複数の等しい大きさのブロックに分割する。ブロックの大きさは、通常8x8画素乃至32x32画素である。

【0022】次に、ブロックごとに、現在のフレームと直前のフレームを比較して、ブロック間の誤差が最小になるベクトルを探索し、そのようなベクトルが発見されたら、直前のフレームにおける当該ブロックの画面上の位置と発見されたベクトルとから現在のフレームにおける当該ブロックの画面上の位置を算出することを反復する。実施例では、誤差を下記の式(1)で定義する。

【0023】

(1)

フレーム(直前フレーム)における座標(x, y)の画素値であ

り、 $Si(x, y)$ は第1フレーム(現在フレーム)における座標 $(x, y)$ の画素値である。 $**2$ は二乗を意味する。式(1)の $dx, dy$ に様々な値を代入してこれを評価し、評価結果を最小にする2次元ベクトル $(dx, dy)$ をもって、ブロックの動きベクトルとする。 $dx, dy$ に代入する数値の範囲は適当に定めることができる。

【0024】第1フレームでは、ブロック $Bj$ は $(x+dx, y+dy)$ に移動しているの、座標位置 $(x+dx, y+dy)$ を新しい座標位置 $(x, y)$ として第1フレームと第 $i+1$ フレームの間で同様の処理を施し、それらフレーム間での動きベクトル $(dx, dy)$ を探索する。

【0025】この処理の結果、図5に示すような、画面のxy軸と時間軸 $t$ を持った3次元空間(Spatio-temporal space;以下ではST空間と呼ぶ)における、注目ブロックの注目シーンを通しての動きベクトルの列(線)が得られる。生成された動きベクトルの列のうち、xy平面に投影した動きの存在範囲が小さすぎるもの及び動作時間が短すぎるものは棄却し、有効な動きを表すベクトル列だけを残す。

$$\Sigma\{(xik-1i)**2 + (yik-yi)**2\}$$

【0028】ここで、 $xik, yik$ は、それぞれ第 $i$ フレームにおけるブロック $Bk$ のx座標、y座標である。また、 $xi, yi$ はそれぞれ第 $i$ フレームにおけるもう一つのブロック $Bi$ のx座標、y座標である。フレーム番号 $i$ の存在領域、つまり距離計算を行う範囲は、2つのブロックの動きベクトル列がオーバーラップして存在する区間とする。

【0029】上記式(2)で計算されたブロックの動きベクトル列間の距離に基づき、公知のクラス分析技術を用いることにより、互いの距離が短い、つまり類似の挙動を示すブロックの動きベクトル列を1つのベクトル列にまとめることができる。例えば、図5に示す4つの動きベクトル列は、図6に示す1つのベクトル列にまとめられる。このようにして生成されたベクトル列を、代表動きベクトル列と呼び、代表動きベクトル列を構成する個々のベクトル(フレーム間変位)を、代表動きベクトルと呼ぶことにする。1つの代表動きベクトル列に関して出力されるデータは、動きの開始フレーム番号 $ts$ 、動きの継続フレーム数 $fn$ 、開始フレームでの動きの開始点の座標値 $(x0, y0)$ 、及びその座標値を起点とする終止点

$$\begin{aligned} Ix &= x0, Iy = y0; \\ Its &= ts/Tint, Itl = tl/Tint; \\ Ixk &= \sum dx_i \\ Iyk &= \sum dy_i \end{aligned}$$

である。

【0032】代表動きベクトル列のフレーム間変位の時間間隔は、フレームのサンプリング間隔に一致し、通常は三十分の一秒程度である。しかしながら、動きは徐々に変化する場合が多く、それを知覚する人間側の感覚

\*【0026】なお、物体の中には、1つのシーンの途中から画面に登場するものもある。そのような物体に対処するために、実施例では、 $M(2以上)$ の所定の数)フレーム毎にxy平面上のブロックの分布をチェックして、ブロックの空白領域があれば、そこに新しいブロックを生成する。新しく生成されたブロックを追跡すれば、途中から登場する物体に関する動きベクトル列を生成することができる。

【0027】d) 代表動きベクトル列の生成(図3のステップ34)

一般に、1つの物体は画面で複数のブロックを占めるので、1つの物体に関して複数の動きベクトル列が生成される。そして、同一の物体に関連して生成された動きベクトル列は類似の挙動を示す。そこで、類似の挙動を示すベクトル列を統合して、その物体に関する1つないしは複数の代表動きベクトル列を生成する。そのためには、ブロックの動きベクトル列間の距離を導入する必要がある。実施例では、動きベクトル列間の距離を下記の式で定義する。

(2)

※フレームまでの代表動きベクトルの列 $(dx1, dy1) i = 0, \dots, fn-1$ 、である。以後、 $ts$ と $fn$ は時間としての概念 $ts$ (開始時刻)と $tl$ (動きの継続時間)に置き換えて説明する。

【0030】なお、1つの物体の動きから、複数の代表動きベクトル列が得られることがあることに注意された。例えば、図7を使って模式的に説明すると、物体Pが破線で示す位置から実線で示す位置まで180度回転した場合には、2つの代表動きベクトル列AとBが得られる。

【0031】e) 動きインデックスの生成及び登録(図3のステップ35、36)

次に、実施例では、後述の検索処理の効率化のため、フレームのサンプリング間隔よりも長い時間間隔 $Tint$ で代表動きベクトル列のデータをサンプリングする。その結果生成される粗いベクトル列を動きインデックスと呼ぶことにする。動きインデックスのデータは、以下の3種類のデータよりなる。(1)開始点座標 $(Ix, Iy)$ 、(2)開始時刻及び動きの継続時間 $(Its, Itl)$ 、(3) $Tint$ 時間内の代表動きベクトルの総和の時系列 $(Iv0, Iv1, \dots, Iv_n)$ 、ただし $Ivk = (Ixk, Iyk)$ である。ここで、

(3)

も数十分の一秒という単位では働いていない。したがって、より粗い時間間隔でサンプリングしたデータだけを使用しても、検索という目的からは不都合はなく、かつマッチングの成本と低減にもなる。そこで、実施例では、図8に模式的に示すように、シーンごとに、シーン

番号、シーンの開始フレーム番号、及びそのシーンから抽出された動きインデックスを、動き情報データベース21に登録する。なお、動き情報データベース21が具体的にどのような媒体を使う記憶装置上で実現されるかは、設計事項にすぎないので、説明を省略する。

#### 【0033】 2. 動きを条件とした検索

以下では、図2のステップ23、24、25及び26の詳細を、図9に示す処理の流れにしたがって説明する。

##### a) 検索条件の入力(ステップ91)

人間が見る実際の動画は2次元のものであり、また計算機が処理できる動画も2次元情報である。そこで、実施例では、検索条件の入力のために、図10に示すように、ディスプレイ画面100上に2次元のパネル101領域を設定し、利用者に適当なデバイスを動かしてそこに所望の動きを描いてもらうことにする。本発明の場合、動きはベクトルとして表現しているため、利用者はパネル101上に線を描くことになる。パネル101にはカーソル(図示せず)を表示し、デバイスの動きに合わせてカーソルの位置をパネル101上にエコーバックする。計算機側では、パネル101上のカーソルの、移動開始点の座標と、移動中の一定の時間ごとの座標をセンサする。この結果、動作時間と動作中の一定時間おきの座標値が条件として入力される。

【0034】 b) 検出による検索条件解釈結果の表示及び編集(ステップ92)

デバイスを動かして検索条件を入力する場合、利用者が考えている条件と機械が解釈した検索条件に差が生じることがある。また、複数の検索条件を入力する場合には、2つ以上の時系列ベクトル間の時間的な関係(前後関係等)を考慮する必要がある。複数の条件を同時に指定することは難しい。したがって、個々の条件を入力した後に、その時間的な前後関係や時間の長さの修正等をインタフェースを介して行い、その結果をプレイバックして確認するという対話編集機能が必要となる。図10及び図11に対話編集機能の例を挙げる。ここでは、動きがスクロールバー110によって制御され、その先端位置が動きの開始時刻を示し、その長さが動きの継続時間を示している。したがって、スクロールバー相互の位置関係は対応する条件(動き)の時間的な前後関係を示す。このスクロールバー110の長さを変更したり位置を動かすことによって、利用者は、検索条件であるところの動きの動作時間の長さ、つまり動きの速度を修正したり、動き同士の間との前後関係を表現したりすることが\*

$$V = (Isx, Isy, Its, Itl, M), M = (Iv0, Iv1, \dots, IvN) \quad (4)$$

ここで Isx, Isy は開始点座標、Its, Itl は開始時刻及び動きの継続時間、(Iv0, Iv1, ..., IvN, ...) であり、ただし Ivk = (Irk, Iyk) は Tint 時間内の変位である。

【0039】 簡単なために、検索用シーンに含まれる動きが一つの場合を例にとって説明する。以下の各式において、検索用シーンの持つ動きに関するデータは先頭に

\*できる。

【0035】 具体的には、ボタン111、112はスクロールバー110の位置を動かすボタンであり、ボタン111をビックすれば、スクロールバー110を所定単位時間分前に動かすことができ、ボタン112をビックすれば、スクロールバー110を所定単位時間分後に動かすことができる。また、ボタン113、114は、スクロールバー110の長さを変えるボタンであり、ボタン113をビックすれば、スクロールバー110を前に

10 所定単位時間分延滞することができ、ボタン114をビックすれば、スクロールバー110を後に所定単位時間分延滞することができる。ディスプレイ画面100上でスクロールバー110の上には、"Play S" ボタン106と"Play A" ボタン107がある。ボタン106をビックすれば、現在編集中の動きがパネル101上に線104として再生される。具体的には、物体を象徴する小円103が、修正された速さで動き、その軌跡が線102として表示される。また、ボタン107をビックすれば、現在編集中の動きのすべてが、編集されたおりの速さ及び前後

20 関係でパネル101上に再生される。その他、編集用ボタンとして、入力された動きをキャンセルするための"Reset" ボタン104、動きを追加するための"Add" ボタン105、編集作業を終了するための"Exit" ボタン108が用意されている。利用者は、これらのボタンを使った編集操作の繰り返しによって、パネル101上に疑似的な動画であるところの検索用シーンを作り上げる。

【0036】 上述の入力及び対話編集の結果、パネル101上に作られた疑似的な動画像(検索用シーン)の動きを表す、開始時刻、動作時間及び時系列の座標ベクトルが得られる。それらデータを時間間隔Tintでサンプリングすることによって、検索用動きデータを生成する。

##### 【0037】 c) マッチング処理(ステップ93)

ステップ92までの処理で、検索用シーンの動きに関する記述は、動画データデータベース22に蓄積されたシーン中の動きのそれと同じ形のものとなる。本ステップでは、それらを用いて、出力候補を決定する。以下、そのための手順を説明する。

【0038】 蓄積されたあるシーンの持つ動きインデックス集合を(SV0, SV1, ..., ...), 検索用シーンの持つ動きインデックス集合を(QV0, QV1, ...)とする。各々の動きインデックス V は、下記のデータの集合である。

40 【0040】 図10に示す検索インタフェースで作成さ

Q を付け、蓄積シーンの動きに関するデータはSを付けて表す。 (例、検索側の開始点座標: QIsx, QIsy) 以下では、検索用シーンに含まれる動きを検索側動きと呼び、蓄積されたシーンに含まれる動きを蓄積側動きと呼ぶことにする。



れた動きのデータは、利用者が“時間tの間にあるものがこの場所からこのように動いた”ということをもベクトル列の形で表現したものと考えられる。この中で提示されている制約条件は、(1)時間(時間t軸上の位置)、(2)位置(xy平面上の位置)、(3)動き(ベクトル列)の3種類である。検索側動きと蓄積側動きの距離dが、以下に示す手法にしたがって計算され、その結果に基づいて、\*

$$d = \sum \sqrt{|(S1sx+S1yk)-(Q1sx+Q1yk)|^2 + |(S1sy+Q1yk)-(Q1sy+Q1yk)|^2} \quad (5)$$

【0042】S1t1 < Q1t1 または S1t1 < Q1t1 のときの式(5)の左辺の定義は、(i) その場で察却する(d=無限大) (ii) 時間的にオーバーラップしている部分のみに対して計算する等が考えられる。どの定義を採用するかは応用例による。

【0043】(ii) 位置の制約条件が利用者に意識されていない場合

例えば、利用者が図13の動きJと動きKを区別しないで検索したいときである。上記式(5)において位置を決定する要因である蓄積側動き及び検索側動きのそれぞれのIsx, Isyを、ST空間の原点に持ってくることによって、純粋に動き(ベクトル列)のみの比較を行う(図14参照)。この場合、式(5)においてS1sx, S1sy, Q1sx, Q1sy

$$\begin{aligned} Q1x' &= F(Q1x0, Q1x1, \dots, Q1xj) \\ Q1y' &= F(Q1y0, Q1y1, \dots, Q1yjl) \\ (k' = 0, 1, \dots, S1tl, j = 0, \dots, Q1tl) \end{aligned} \quad (6)$$

である。このようにして、Q1tlをS1tlに合わせれば、距離dは上記の式(5)で計算区間をS1tlにして算出できる。

【0045】(iv) 位置と時間に関する制約条件がどちらも意識されていない場合

図15に示すように、位置と時間の両方に関して、検索側動きと蓄積側動きに合わせる。

【0046】以上のように、検索条件における位置や時間に関しての曖昧さが存在する場合は、ST空間でのベクトル列(線)の変形操作によって検索条件と動きインデックスを比較可能な状態にし、距離を求める。この距離計算を全ての蓄積されたベクトル列に対して行ない、距離の小さなものから順にそのベクトル列を持つシーンを出力候補とする。

【0047】検索条件の曖昧さは、具体的事例によってその度合が異なると考えられる。そこで、曖昧さを検索時に隔に指定するインタフェースを用意することが好ま

$$\begin{aligned} [1] (Q0, S0) & [2] (Q0, S1) & [3] (Q0, S2) & [4] (Q0, S3) \\ (Q1, S1) & (Q1, S0) & (Q1, S1) & (Q1, S0) \\ [5] (Q0, S0) & [6] (Q0, S1) & [7] (Q0, S2) & [8] (Q0, S3) \end{aligned}$$

\*両者の類否が判定される。

【0041】(i) 位置や時間に関する制約条件が利用者に意識されていない場合

Q1ts及びS1tsをST空間上の時間軸の原点に移動して、下記の式(5)にしたがって距離dを計算する(図12参照)。

【数1】

※syの項を取り去ってdを計算する。

【0044】(iii) 時間(動きの速度)に関する制約条件が意識されていない場合

2つの動きを時間に関して整合させる。すなわち、S1tlとQ1tlの間での時間合わせを行う。この場合、単純に時間を線形に変化させるのであれば、Q1tlの値をS1tlに一致するように伸長もしくは短縮し、それに合わせてベクトル列Q1vk=(Q1xk, Q1yk)からベクトル列Q1vk'=(Q1xk', Q1yk')を作り直す。また、非線形に時間の変形を許すのであれば、公知のD Pマッピング等の方法を用いて、ベクトル列Q1vk'=(Q1xk', Q1yk')をつくる。変形画像をFとすると、

しい。例えば、検索条件の時間または位置が不定であることを、利用者がメニューで指示できるようにし、システム側はその指示に応じて、上記(1)乃至(iv)に示した手法のいずれかで距離計算を行う。

【0048】複数の検索条件を入力した場合も、上記と同様の距離計算をすることによって、蓄積側動きと検索側動きの対応付けられた組ごとに、距離尺度を設けることができる。ただし、この場合、組合せの数が増えることを考慮しなければならない。例えば、図16に示す例では、検索用シーンには、動きQ0, Q1が含まれ、蓄積されたシーンには動きS1, S2, S3が含まれている。何の条件もつけないなら、検索側の2つの動きと蓄積側の2つの動きの組合せは下記のように12通り考えられ、したがって距離計算を24回実行しなければならない。

【0049】

13				14	
(Q1, S2)	(Q1, S2)	(Q1, S1)	(Q1, S1)		
[9] (Q0, S0)	[10] (Q0, S1)	[11] (Q0, S2)	[12] (Q0, S3)		
(Q1, S3)	(Q1, S3)	(Q1, S3)	(Q1, S2)		

【0050】そこで、検索側動き間の時間的前後関係や位置の情報を利用して、組合せの数を絞りこむ。まず、時間的前後関係の利用について説明する。実施例では、時間に関しての前後関係を記述する索引ファイルを、予め同一のシーンにある動きインデックスの開始時刻S1sの値に基づいて作成しておく。図17に例示する動きインデックスS0, S1, S2, S3からは、図18のような索引ファイルが作成される。利用者が検索条件相互の時間的

[1] (Q0, S0)	[2] (Q0, S0)	[3] (Q0, S1)	[4] (Q0, S1)
(Q1, S2)	(Q1, S3)	(Q1, S2)	(Q1, S3)

【0052】次に、位置情報の利用について説明する。実施例では、蓄積された動きインデックスの開始点(S1sx, S1sy)が、画面を分割して設定された複数の領域のうちいずれに存在するかを記述する索引ファイルを予め作成しておく。例えば、図19に示す動きS0, S1, S2, S3からは、S0が領域R1に、S1とS3が領域R2に、S2が領域R4に関連付けられることを示す索引ファイル(図20)が作成される。ここで、R1, R2, R3, R4は、xy平面を四等分して設定された領域である。

【0053】検索側動きの開始点座標が決定したら、その付近に開始点がある蓄積側動きのみを距離計算の対象とすることによって、検索のコストを削減することができる。図19に示す例において、検索側動きQ0とQ1の位置が意識されているとする。Q0とQ1の開始点の座標(Q1sx, Q1sy)から、Q0が領域R1に関連付けられ、Q1が領域R3に関連付けられることがわかる。したがって、索引ファイル(図20)を調べると、Q0と開始点に近い蓄積側動きはS0であり、Q1と開始点に近い蓄積側動きは、S1とS3であることがわかる。よって、距離計算の対象になる検索側動きと蓄積側動きの組合せは、下記の2通りに絞り込まれる。

【0054】

[1] (Q0, S0)	[2] (Q0, S0)
(Q1, S1)	(Q1, S3)

【0055】以上のようにして絞りこんだ組合せの各々について、対応づけられたベクトル間の距離dを算出し、その和を求めると。そのようにして計算した和の最も小さなものを、そのシーンを代表する距離とする。全てのシーンに対して距離を計算した後、最も小さなものから順に出力候補とする。

【0056】d) 画像格納アドレスの生成(図9のステップ94)

対照表を使う等の、既述の方法によって、出力候補のシーンの開始フレーム番号を、動画データベース2における当該フレーム画像の格納アドレスに変換する。

\*前後関係を意識している場合には、その索引ファイルを使って、検索条件と同じ時間的前後関係を有する、蓄積側の動きインデックスの組を発見する。例えば、図17に示す検索側動きQ0とQ1の前後関係が意識されているとする。前後関係を有する蓄積側動きインデックスの組(対)は、図18の索引ファイルから4組であることがわかる。よって、距離計算の対象になる検索側動きと蓄積側動きの組合せは、以下に示す4通りに絞り込まれる。

【0051】

(Q0, S1)	[4] (Q0, S1)
(Q1, S2)	(Q1, S3)

【0057】e) 候補シーンの表示(ステップ95)

ステップ93で計算した距離の小さなシーンから順に、ステップ94で得られた格納アドレスをアクセスして、その開始フレームの画像データを表示装置(図示せず)に転送・表示する。利用者は、その開始フレーム(静止画像)を見て、所望のシーンであるか否かを判断する。動画画像を見たいときは、後続フレームを順次再生すればよい。

【0058】3. 変形例

以上、本発明を好適な実施例に即して説明したけれども、本発明の適用可能な範囲はこれに限られない。まず、動きを入力するデバイスとして使用可能なものは、マウスに限られない。その空間的な位置を変化させることにより利用者が動きを表現できるものであれば何でもよく、例えばデータ・グローブを使ってもよい。

【0059】シーンの検索において、従来のキーワード検索を併用することも可能である。その場合、シーンに付与された文字・数値ばかりでなく、検索しようとする動きを持つ物体の、名前、色、形等の属性を指定してもよい。もちろん、物体の色を言葉で指定する代りに、色のメニューから選択してもよい。また、物体の形を指定するのに、言葉ではなく、マウス等で輪郭を描いて指定するようにしてもよい。

【0060】上記実施例では、検索の効率の観点から、代表動きベクトル列は捨てて、動きを粗くサンプリングした動きインデックスのみを動き情報データベースに登録した。しかし、アプリケーションによっては、代表動きベクトル列のレベルでのマッチングが必要となることもある。そのような場合には、代表動きベクトル列と動きインデックスとを動き情報データベースに登録しておき、動きインデックスのレベルで粗くマッチングを実行し、マッチした候補についてさらに代表動きベクトル列のレベルでマッチングを実行して、出力候補を決定すればよい。もちろん、動きインデックスを生成せず、代表動きベクトル列だけを動き情報データベースに登録し、代表動きベクトル列レベルのマッチングだけを

実行することは可能である。しかし、一般的に、検索コストの点で、実施例の方式の方が有利である。

【0061】また、上記実施例では、動き情報データベースに、物体の動きを表わす時系列のベクトルを登録した。しかしながら、時系列のベクトルに代えて、物体の位置の座標の時系列を登録してもよい。その場合、検索条件が座標の時系列として入力されるなら、式(5)で定義される距離に基づくマッチングを行うことができる。なぜなら、式(5)の中で括弧でくくられた4つの項は、x

y空間の座標を表わしているからである。しかしながら、検索条件の曖昧さに柔軟に対処するためには、実施例のように、動き情報をベクトルの時系列の形式で蓄積する方が有利である。

【0062】また、図19及び図20を使って説明した例では、画面を等しい大きさのブロックR1乃至R4に分割して索引を作成したけれども、公知のkd-tree等の手法を用いて、画面を蓄積データの分布に応じた可変ブロックに分割して索引付けを行うことも可能である。

【0063】

【発明の効果】以上、本発明は、人間にとって曖昧な動きという情報を、デバイスの空間的な位置が変化させることによって、頭に思い描いたままに近い形で入力することによって、動画像の一部分を検索する環境を提供するものである。また、そのような検索の前提となる、画面上に登場する物体の動き情報の自動抽出及び抽出された情報の動き情報のデータベースへの蓄積方法、入力された条件と動き情報データベースとのマッチングの方法を併せて提供することによって、動きを条件とした、利用者に優しく効率のよい動画像の検索を可能とするものである。

【図面の簡単な説明】

【図1】従来の動画像検索技術における処理とデータの流れを示すブロック図である。

【図2】本発明における処理とデータの流れを示すブロック図である。

【図3】動き情報抽出処理の流れを示す図である。

【図4】動きベクトル生成処理におけるブロック及びその動きを説明する図である。

【図5】ST空間におけるブロックの動きベクトル列を示

す図である。

【図6】ST空間における代表動きベクトル列を示す図である。

【図7】回転物体から生成される代表動きベクトル列の例を示す図である。

【図8】動き情報データベースの構成を説明する図である。

【図9】検索処理とデータの流れを示すブロック図である。

【図10】検索条件の編集画面を示す図である。

【図11】検索条件の編集画面の一部を示す図である。

【図12】ST空間で蓄積された動きインデックスと検索用インデックスの距離の定義の一例を説明する図である。

【図13】2つの動きのXY平面への投影図である。

【図14】ST空間で蓄積された動きインデックスと検索用インデックスの距離の定義の他の例を説明する図である。

【図15】ST空間で蓄積された動きインデックスと検索用インデックスの距離の定義のさらに他の例を説明する図である。

【図16】検索用動きインデックスが複数個ある場合の蓄積動きインデックスとの対応づけを説明するための図である。

【図17】検索用動きインデックスが複数個あり、かつそれらの時間的前後関係を考慮に入れる場合の、蓄積動きインデックスとの対応づけを説明するための図である。

【図18】蓄積動きインデックスの時間的前後関係を蓄積する索引ファイルの構成を示す図である。

【図19】検索用動きインデックスが複数個あり、かつそれらの位置関係を考慮に入れる場合の、蓄積動きインデックスとの対応づけを説明するための図である。

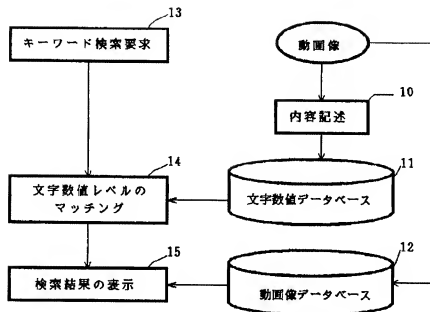
【図20】蓄積動きインデックスの位置関係を蓄積する索引ファイルの構成を示す図である。

【符号の説明】

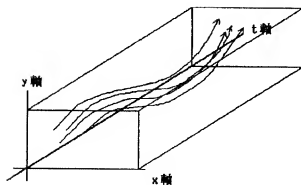
20 動き情報データベース、 21 動画像データベース

【図1】

(従来技術)

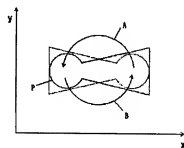


【図5】

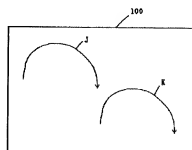


ブロックの動き

【図7】

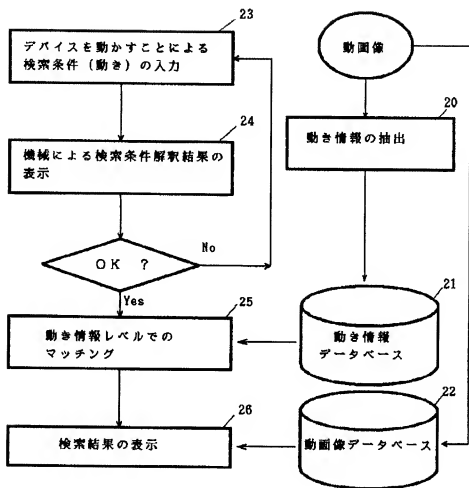


【図13】

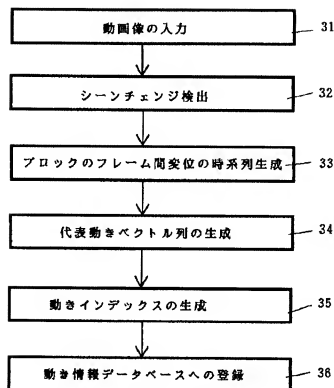


【図2】

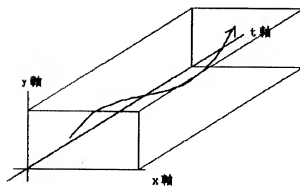
## ( 発 明 )



【図3】

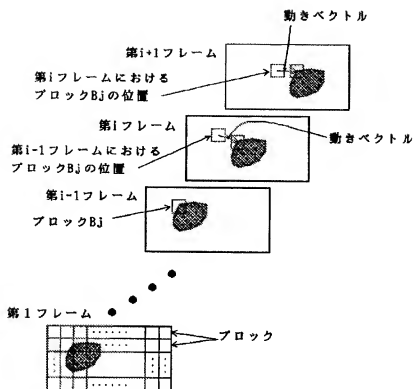


【図6】

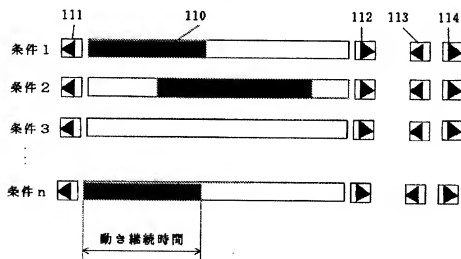


統合された動き

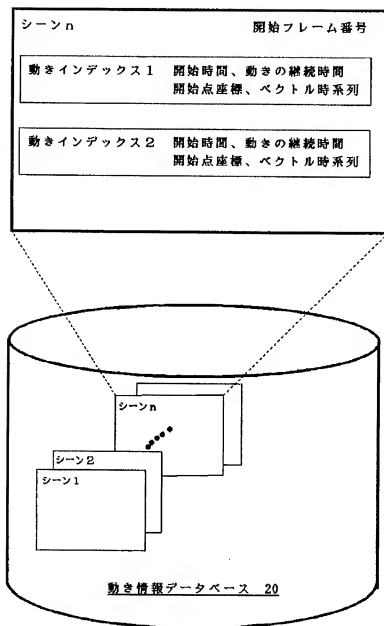
【図4】



【図11】

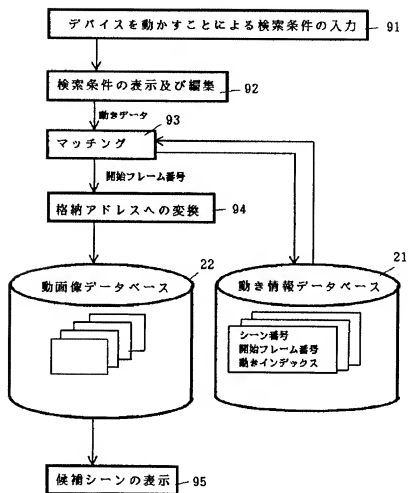


【図8】

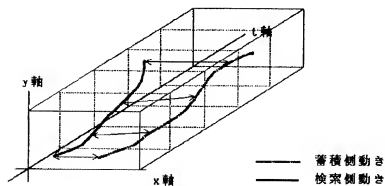




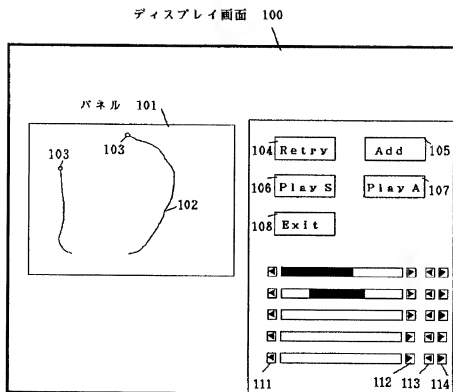
【図9】



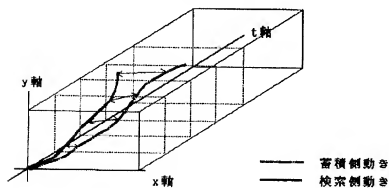
【図12】

距離  $d$  の定義

【図10】

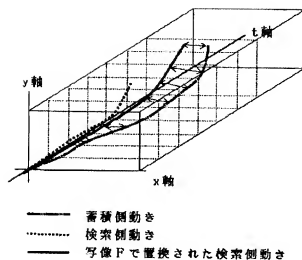


【図14】



位置不定の場合の距離  $d$

【図15】



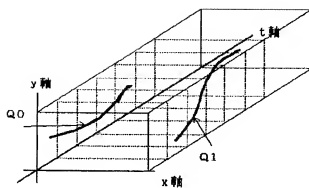
位置不定かつ時間不定の場合の距離  $d$

【図18】

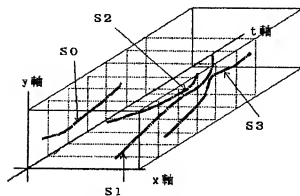
シーン番号	前後関係
• • • •	
シーン N	$S0 = S1 < S2 = S3$
• • • •	

索引ファイル

[図16]



検索側動き

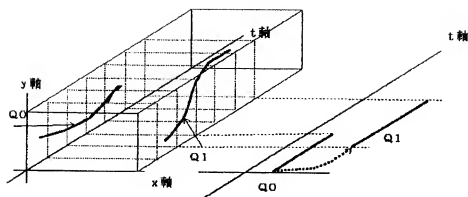


蓄積側動き

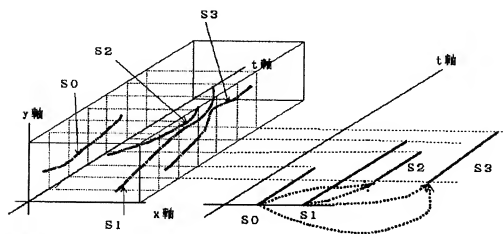
[図20]

シーン番号	R 1	R 2	R 3	R 4
⋮				
シーン N	S0		S1, S3	S2
⋮				

【図17】



検索側動き



蓄積側動き

【図19】

